Amendments to the Claims:

This listing of claims will replace all prior versions, and listings, of claims in the application:

Listing of Claims:

1. (Currently Amended) A method for monitoring multiple online resources in different formats, the method comprising the steps of:

identifying an online resource to monitor, the online resource being stored in a first format, the online resource in the first format including data in a non-strict architectural structure;

converting the online resource to a strict formatted file, wherein data in the first format of the online resource is converted into a strict architectural structure in the strict formatted file;

identifying relevant data based on the strict architectural structure of the data in the strict formatted file using an analytic parser; and determining whether the identified relevant data has been altered.

2. (Previously Presented) The method of claim 1 wherein the online resource is a HyperText Markup Language application.

3. (Previously Presented) The method of claim 1 wherein the online resource is a non-HyperText Markup Language application.

4. (Previously Presented) The method of claim 3 further comprising the step of converting the online resource from the non-HyperText Markup Language application to a HyperText Markup Language application, wherein converting the online resource to the strict formatted file comprises converting the HyperText Markup Language application to the strict formatted file.

5. (Previously Presented) The method of claim 1 wherein an Extensible Style Sheet Transform is used to convert the online resource to the strict formatted file.

6. (Previously Presented) The method of claim 1 wherein the strict formatted file is an Extensible Markup Language application.

7. (Previously Presented) The method of claim 1 wherein the strict formatted file is an Extensible HyperText Markup Language application.

8. (Previously Presented) The method of claim 1 wherein the strict formatted file is a document object model of the online resource.

9. (Previously Presented) The method of claim 1 wherein the analytic parser is a script that operates on the strict formatted file.

10. (Previously Presented) The method of claim 9 wherein the script identifies relevant data via markers within the strict formatted file.

11. (Previously Presented) The method of claim 1 wherein an altered file is determined by comparing the identified relevant data to a most recent archived copy of the identified relevant data.

12. (Previously Presented) The method of claim 11 further comprising the step of storing the identified relevant data within a database.

13. (Previously Presented) The method of claim 1 further comprising the step of automatically notifying a user when the identified relevant data has changed.

14. (Previously Presented) The method of claim 1 further comprising the step of automatically updating a database.

15. (Currently Amended) A system for monitoring multiple files in disparate formats, the system comprising:

a file type identifier module adapted to identify the format of a particular online resource, the online resource in the first format including data in a non-strict architectural structure;

a format conversion module adapted to convert the online resource to a strict formatted file, wherein data in the format of the online resource is converted into a strict architectural structure in the strict formatted file;

an analytic parser adapted to identify relevant data in the strict architectural structure in ~~within~~ the strict formatted file;

a resource filter adapted to determine whether the identified relevant data has been altered.

16. (Previously Presented) The system of claim 15 wherein the online resource is a HyperText Markup Language application.

17. (Previously Presented) The system of claim 15 wherein the online resource is a non-HyperText Markup Language application.

18. (Currently Amended) The system of claim ~~15~~ 17 further comprising an HTML conversion module adapted to convert the online resource from the non-HyperText Markup Language application to a HyperText Markup Language application, wherein the format conversion module is adapted to convert the online resource to the strict formatted file by converting the HyperText Markup Language application to the strict formatted file.

19. (Previously Presented) The system of claim 15 wherein an Extensible Style Sheet Transform is used to convert the online resource to the strict formatted file.

20. (Previously Presented) The system of claim 15 wherein the strict formatted file is an Extensible Markup Language application.

21. (Previously Presented) The system of claim 15 wherein the strict formatted file is an Extensible HyperText Markup Language application.

22. (Previously Presented) The system of claim 15 wherein the strict formatted file is a document object model of the online resource.

23. (Previously Presented) The system of claim 15 wherein the analytic parser is a script that operates on the strict formatted file.

24. (Previously Presented) The system of claim 23 wherein the script identifies relevant data via markers within the strict formatted file.

25. (Previously Presented) The system of claim 15 wherein an altered file is determined by comparing the identified relevant data to a most recent archived copy of the identified relevant data.

26. (Previously Presented) The system of claim 15 wherein the identified relevant data is stored within a database.

27. (Previously Presented) The system of claim 15 further comprising a monitoring module adapted to automatically notify a user when the identified relevant data has changed.

28. (Previously Presented) The system of claim 15 further comprising a monitoring module adapted to automatically update a database when the identified relevant data has changed.

29. (Currently Amended) A method for monitoring multiple online resources in different formats, the method comprising the steps of:

identifying an online resource to monitor, the online resource being stored in a first format, the online resource in the first format including data in a non-strict architectural structure;

converting the online resource to a strict formatted file, wherein data in the first format of the online resource is converted into a strict architectural structure in the strict formatted file;

identifying relevant data based on the strict architectural structure in the strict formatted file using analytic parser; and remotely updating the relevant data using a script.

30. (Currently Amended) A system for monitoring multiple files in disparate formats, the system comprising:

a file type identifier module adapted to identify the format of a particular online resource, the online resource in the first format including data in a non-strict architectural structure;

a format conversion module adapted to convert the online resource to a strict formatted file, wherein data in the format of the online resource is converted into a strict architectural structure in the strict formatted file;

an analytic parser adapted to identify relevant data in the strict architectural structure in ~~within~~ the strict formatted file; and

a resource updater to update the identified relevant data.

31. (new) The method of claim 1, wherein identifying relevant data in the strict formatted file comprises identifying data flags or identifiers in the strict architectural structure to identify the relevant data.

32. (new) The system of claim 15, wherein the analytic parser is adapted to identify data flags or identifiers in the strict architectural structure to identify the relevant data.

33. (new) The method of claim 29, wherein identifying relevant data in the strict formatted file comprises identifying data flags or identifiers in the strict architectural structure to identify the relevant data.

34. (new) The system of claim 30, wherein the analytic parser is adapted to identify data flags or identifiers in the strict architectural structure to identify the relevant data.